AMENDMENTS TO THE CLAIMS

Please amend the claims as follows.

1. (Currently Amended) A method for converting a source program into a plurality of tokens on a computer system, comprising:

obtaining one or more entries;

analyzing said source program using said computer system; and

generating said plurality of tokens from said source program at runtime using said computer system,

wherein said entries may be used to generate a subset of said plurality of tokens, and wherein at least one of said entries is added at runtime.

2. (Previously Presented) The method of claim 1 wherein said entries comprise a language descriptor and a token value.

3. (Original) The method of claim 2 wherein the analyzing comprises:

obtaining a lexeme from said source program; and

determining if said lexeme matches one of said language descriptors.

4. (Original) The method of claim 3 wherein the analyzing further comprises:

obtaining said token value if said lexeme matches one of said language descriptors.

5. (Original) The method of claim 4 wherein the analyzing further comprises:

obtaining a next lexeme from said source program.

6. (Previously Presented) The method of claim 5 wherein the generating comprises:

outputting said token value in response to a request from a host program.

7. (Original) The method of claim 6 wherein said language descriptor is a reserved word.

8. (Original) The method of claim 6 wherein said language descriptor is an operator.

9. (Previously Presented) The method of claim 1 wherein the obtaining further comprises:

entering said entries into a token dictionary.

10. (Currently Amended) A computer program product comprising:

> a computer usable medium having computer readable program code embodied therein configured to convert source program into a plurality of tokens, said computer program product comprising:
>
> > computer readable code configured to cause a computer to obtain one or more entries;
> >
> > computer readable code configured to cause a computer to analyze said source program ; and
> >
> > computer readable code configured to cause a computer to generate said plurality of tokens from said source program at runtime,
> >
> > wherein said entries may be used to generate a subset of said plurality of tokens, and
> >
> > wherein at least one of said entries is added at runtime.

11. (Original) The computer program product of claim 10 wherein said entries comprise a language descriptor and a token value.

12. (Original) The computer program product of claim 11 wherein said computer code configured to cause a computer to analyze the source program comprises:

> computer readable code configured to cause a computer to determine if said lexeme matches one of said language descriptors.

13. (Original) The computer program product of claim 12 wherein said computer code configured to cause a computer to analyze said source program further comprises:

> computer readable code configured to cause a computer to obtain said token value if said lexeme matches one of said language descriptors.

14. (Original) The computer program product of claim 13 wherein said computer code configured to cause a computer to analyze said source program further comprises:

> computer readable code configured to cause a computer to obtain a next lexeme from said source program.

15. (Previously Presented) The computer program product of claim 14 wherein said computer code configured to cause a computer to generate said plurality of tokens comprises:

    computer readable code configured to cause a computer to output said token value in response to a request from a host program.

16. (Original) The computer program product of claim 15 wherein said language descriptor is a reserved word.

17. (Original) The computer program product of claim 15 wherein said language descriptor is an operator.

18. (Previously Presented) The computer program product of claim 10 wherein said computer code configured to cause a computer to obtain one or more entries further comprises:

    computer readable code configured to cause a computer to enter said entries into a token dictionary.

19. (Currently Amended) A lexical analyzer, comprising:

    one or more entries configured to be obtained;

    a source program analyzer;

    a plurality of tokens configured to be generated at runtime from said source program analyzer,

    wherein said entries may be used to generate a subset of said plurality of tokens, and wherein at least one of said entries is added at runtime.

20. (Original) The lexical analyzer of claim 19, wherein said entries comprise a language descriptor and a token value.

21. (Previously Presented) The lexical analyzer of claim 20 wherein said source program analyzer comprises:

    a source program interface, wherein said source program interface obtains a lexeme from a source program; and

    a lexeme comparator, wherein said lexeme comparator determines whether said lexeme matches one of said language descriptors.

22. (Original) The lexical analyzer of claim 21, wherein said source program analyzer further comprises:

a token output interface, wherein said interface generates said token if said lexeme matches one of said language descriptors.

23. (Original) The lexical analyzer of claim 22, wherein said source program interface further comprises:

a source program manager, wherein said manager obtains a next lexeme from said source program.

24. (Original) The lexical analyzer of claim 23, wherein said output interface comprises:

a host program event handler, wherein said event handler causes said output interface to generate said token value in response to a request from the host program.

25. (Original) The lexical analyzer of claim 24, wherein said language descriptor is a reserved word.

26. (Original) The lexical analyzer of claim 24, wherein said language descriptor is an operator.

27. (Original) The lexical analyzer of claim 19, further comprising:

a token dictionary, wherein said entries comprise dictionary entries.